LAYERED INSIGHTS INTO PYRAMID FEATURE FUSION ARCHITECTURE FOR SSL

Sourabh Prakash, Ashrya Agrawal & Priyanshi Shah *
University of California, San DIego
La Jolla, CA 92092, USA
{soprakash,asagrawal,prs003}@ucsd.edu

ABSTRACT

In this paper, we introduce a novel self-supervised learning(SSL) approach leveraging pyramid layers to extract essential visual features. Employing image inpainting as the pretext task, we empirically demonstrate the effectiveness of this methodology by rigorously evaluating the trained network's performance in downstream image classification/segmentation tasks. Our findings underscore the substantial performance improvement achieved through SSL. Furthermore, we study the individual contributions of each self-supervised trained feature pyramid layer to the model's performance enhancement. Findings here, led us to conclude that optimal number and size of feature pyramid layers vary for each model and significantly influence overall performance.

1 INTRODUCTION

At its core, any self-supervised technique depends heavily on the model architecture. In this paper: 1) We propose a new network architecture for SSL and, 2) Do an indepth study on the contribution of each layer to the model performance.

The architecture is based on pyramid layers and uses the encoder-decoder layers at every scale to extract the visual features using SSL. A key challenge is the optimal layer configuration in the pyramid structure. Our findings underscore this: 1) Removing the last layer increases classification accuracy from 28.55% to 29.59%, and 2) Eliminating the first layer improves segmentation IOU from 0.4901 to 0.4982. These results highlight the critical role of layer configuration in SSL performance.

	Model	**Full Pyramid**	**Layer-1**	**Layer-2**	**Layer-3**
Classification(acc.)	SSL	28.5575	23.9143	29.1592	29.5983
	Non SSL	24.2478	25.2074	23.3534	23.1094
Segmentation(IOU)	SSL	0.4901	0.4982	0.4931	0.4968
	Non SSL	0.483	0.4911	0.4822	0.4871

Table 1: Comparison of classification accuracy with and without SSL for different layers removed

2 METHODS

Inpainting Foundation for Downstream Image Tasks. For the pretext task(image inpainting), we utilize the CIFAR-10 dataset Krizhevsky (2009). To assess the network's performance in downstream classification and segmentation tasks, we employ the Flowers102 Nilsback & Zisserman (2008) and PASCAL VOC 2007 Everingham et al. (2007) datasets, respectively. In the inpainting task, a random patch is removed from an input image, and the encoder-decoder network is tasked with generating this missing patch while maintaining the overall coherence of the entire image Pathak et al. (2016).To successfully accomplish this, the network must first contextualize the

*Equal contribution.

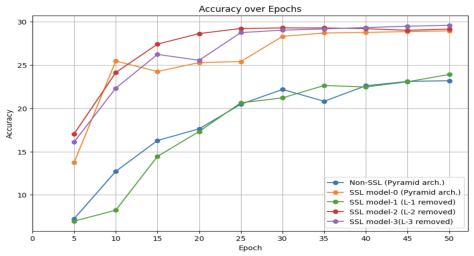


Figure 1: Classification accuracy

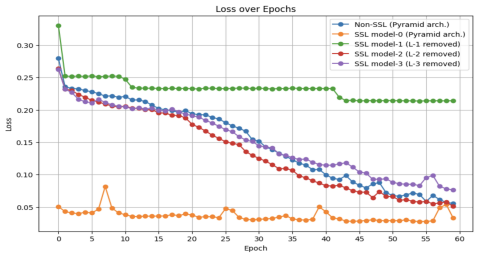


Figure 2: Segmentation loss over the epochs

remainder of the image, thereby necessitating the extraction of pertinent features from the input image. The varying resolutions provided by each layer in the pyramid network enable the extraction of multi-scale features, thereby enhancing the overall learning from of the inpainting task.

We assess the self-supervised model using two downstream tasks: Image Classification and Image Segmentation. For Image Classification Evaluation, we optimize by consolidating encoders from various pyramid layers and replacing decoders with fully-connected layer while subsequently fine-tuning the model. Next we analyze the contribution of each layer of the pyramid to the predictive power of SSL model. We systematically remove feature extractors generated by the encoder of each layer, while keeping the corresponding components from the remaining two layers intact. Then, we fine-tune the SSL model equipped with a classification head and evaluate its performance.

3 Results and Discussions

The pyramid SSL architecture for the classification task results in an accuracy of 28.55%, 4% better performance than Non-SSL model(24.24%). While for the segmentation task the increase the SSL model results in IOU of 0.4901 vs 0.483 of the Non-SSL architecture.

Our findings, illustrated in 1, demonstrate that the exclusion of the first (largest resolution), second (intermediate resolution), and third (smallest resolution) layers result in classification accuracy of 23.91%, 29.15%, and 29.59%, respectively. Interestingly, the combination of the second and third layers, in the absence of the first, yielded performance comparable to the Non-SSL model (24.24% accuracy), suggesting that these two layers compensate for the information loss (approximately 75%) incurred by the absence of the largest layer.

Upon removing the first layer, in classification task as shown in Figure 1 we see major drop in performance whereas in the segmentation task while loss decreases in Figure 2, the IoU value improves from 0.4901 to 0.4982 for the SSL model. This indicates that the behavior can vary across tasks and we need to perform layer-wise analysis to know the optimal depth. Moreover, the higher accuracy observed upon removing the last layer, as shown in Table1, again in both tasks suggests a critical trade-off in the pyramid structure. This trade-off varies across applications, highlighting the necessity for application-specific architectural optimization.

The motivation for this study stems from the ongoing debate in the ML community on the optimal depth, Mousatov & Hartnoll (2020), Ba & Caruana (2014) and complexity of neural network architectures Lin et al. (2013), for various tasks. Our empirical findings show that more layers do not necessarily result in better performance for self-supervised learning. Such insights contribute to the broader understanding of effective network design in the realm of self-supervised learning.

4 Conclusion

Our research employing pyramid layers in self-supervised learning demonstrates significant improvement in model performance for downstream tasks. The study underscores the critical balance between image scale reduction and information retention, offering key insights for the optimization of self-supervised in computer vision. Future work will focus on extending this analysis to a variety of complex neural network architectures and additional layers, larger datasets, and exploring applications in object detection.

URM STATEMENT

The authors acknowledge that at least one key author of this work meets the URM criteria of ICLR 2024 Tiny Papers Track.

REFERENCES

N. Alosaimi, H. Alhichri, Y. Bazi, et al. Self-supervised learning for remote sensing scene classification under the few shot scenario. *Sci Rep*, 13:433, 2023. doi: 10.1038/s41598-022-27313-5.

Jimmy Ba and Rich Caruana. Do deep nets really need to be deep? *Advances in neural information processing systems*, 27, 2014.

Mariusz Bojarski, Davide Del Testa, Daniel Dworakowski, Bernhard Firner, Beat Flepp, Prasoon Goyal, Lawrence D Jackel, Mathew Monfort, Urs Muller, Jiakai Zhang, et al. End to end learning for self-driving cars. *arXiv preprint arXiv:1604.07316*, 2016.

M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2007 (VOC2007) Results. http://www.pascal-network.org/challenges/VOC/voc2007/workshop/index.html, 2007.

T. Gan, Z. Jin, L. Yu, et al. Self-supervised representation learning using feature pyramid siamese networks for colorectal polyp detection. *Sci Rep*, 13:21655, 2023. doi: 10.1038/s41598-023-49057-6.

L. Huang, H. Chen, T. Liu, et al. Self-supervised learning of hologram reconstruction using physics consistency. *Nat Mach Intell*, 5:895–907, 2023a. doi: 10.1038/s42256-023-00704-7.

Shih-Cheng Huang, Anuj Pareek, Malte Jensen, Matthew P Lungren, Serena Yeung, and Akshay S Chaudhari. Self-supervised learning for medical image classification: a systematic review and implementation guidelines. *NPJ Digital Medicine*, 6(1):74, 2023b.

Alex Krizhevsky. Learning multiple layers of features from tiny images. pp. 32–33, 2009. URL `https://www.cs.toronto.edu/~kriz/learning-features-2009-TR.pdf`.

Min Lin, Qiang Chen, and Shuicheng Yan. Network in network. *arXiv preprint arXiv:1312.4400*, 2013.

Connie H Mousatov and Sean A Hartnoll. On the planckian bound for heat diffusion in insulators. *Nature Physics*, 16(5):579–584, 2020.

Maria-Elena Nilsback and Andrew Zisserman. Automated flower classification over a large number of classes. In *Indian Conference on Computer Vision, Graphics and Image Processing*, Dec 2008.

Deepak Pathak, Philipp Krahenbuhl, Jeff Donahue, Trevor Darrell, and Alexei A. Efros. Context encoders: Feature learning by inpainting, 2016.

A APPENDIX

A.1 DATASETS

Self-supervised learning frequently leverages extensive unlabeled datasets. In our pretext tasks, we utilize the CIFAR10 dataset Krizhevsky (2009), which contains 60,000 32x32 color images, equally divided into 10 classes. To assess the network's performance in downstream classification and segmentation tasks, we employ the Flowers102 Nilsback & Zisserman (2008) and PASCAL VOC 2007 (Everingham et al., 2007) datasets, respectively.

A.2 INPAINTING AS PRETEXT FOR SSL

In the inpainting task, a random patch is removed from an input image, and the encoder-decoder network is tasked with generating this missing patch while maintaining the overall coherence of the entire image. This choice of pretext task is motivated by the fact that the network is presented with the complete image Pathak et al. (2016) and then challenged to predict the missing pixels. To successfully accomplish this, the network must first contextualize the remainder of the image, thereby necessitating the extraction of pertinent features from the input image. The height and width of the patches are half of the image sizes and the patches are placed arbitrarily.

A.3 PYRAMID ARCHITECTURE.

In this work, we propose a novel self-supervised learning framework employing a pyramid structure for multi-scale image representation. Our architecture, depicted in Figure 3, utilizes three resolution layers (1, $\frac{1}{2}$, $\frac{1}{4}$) to address the inpainting task. At each scale, we apply the self-supervised inpainting technique discussed earlier. Each scale consists of an encoder and a decoder. The encoder comprises four convolutional layers, sequentially augmented with ReLU activation functions, commencing with 3 channels and concluding with 512 channels. Subsequently, we employ the decoder to reconstruct the original image, complete with the previously removed patch, using our neural network. The decoder is constructed using convolutional transpose layers that correspond to the convolutional layers in the decoder.

During the reconstruction process, features from each decoder stage are upsampled using bilinear interpolation, followed by element-wise adaptive average pooling. This final reconstructed image serves as the basis for computing the loss function and for facilitating backpropagation. Importantly, our pipeline is trained end-to-end, rather than at individual stages. This design aligns with the multi-scale feature extraction methods shown to be effective in diverse applications such as medical image analysis and remote sensing Gan et al. (2023), Alosaimi et al. (2023).

Baseline: For the scope of this paper, we used a small model and hence the appropriate baseline we choose is the non SSL model which has all the layers as described above. For further details on the architecture and baseline, refer the code made public at https://github.com/Spra09/layered_insights_SSL.

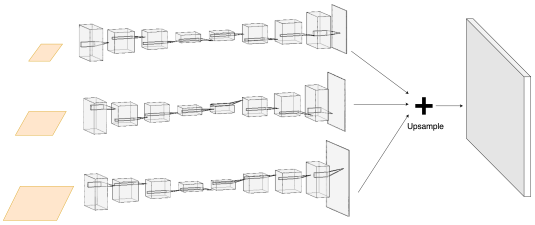


Figure 3: Gaussian pyramid encoder-decoder Architecture in the context of Self Supervised learning

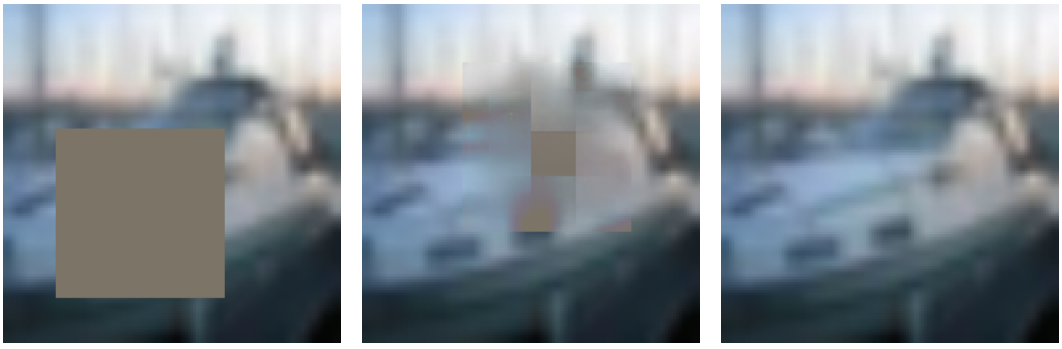
The decoder utilizes convolutional transpose layers corresponding to the encoder's configuration, employing bilinear interpolation and adaptive average pooling for image reconstruction. This approach is inspired by techniques used in complex reconstruction tasks in holography and medical imaging Huang et al. (2023a), Huang et al. (2023b). This reconstructed final image serves as the basis for computing the loss function and for facilitating backpropagation. Importantly, our pipeline is trained end-to-end, mimicking real world behaviour where intermediary signals are absent Bojarski et al. (2016).

A.4 PROOF THAT DEEPER NEURAL NETWORKS CAN RESULT IN POOR PERFORMANCE

Consider a semi-supervised learning setting with a labeled dataset $D_l = \{(x_i, y_i)\}_{i=1}^{l}$ and an unlabeled dataset $D_u = \{x_j\}_{j=1}^{u}$. Let $f(x;\theta)$ be a neural network parameterized by θ. This section aims to show that as the depth of the network increases, the risk of overfitting increases.

	Model	Full Pyramid	Layer-1	Layer-2	Layer-3
Segmentation(acc.)	SSL	0.9312	0.9447	0.9303	0.9299
	Non SSL	0.9287	0.9447	0.9333	0.9311

Table 2: Comparison of classification accuracy with and without SSL for different layers removed



(a) Image with a Patch (b) Completed Image (c) Ground Truth

Figure 4: Inpainting SSL Task Images. Figure b) shows the results by our pre-trained model. The results indicate that the model performs well and can generalize well for the downstream task

A.4.1 PRELIMINARIES

VC Dimension: The Vapnik-Chervonenkis (VC) dimension is a measure of the capacity of a set of functions. For a class of functions $\mathcal{F}$, the VC dimension, denoted as VCdim($\mathcal{F}$), is the maximum number of points that can be shattered by $\mathcal{F}$.

Empirical Risk Minimization: Given a hypothesis class $\mathcal{H}$, empirical risk minimization seeks to find $h \in \mathcal{H}$ that minimizes the empirical risk.

A.4.2 PROOF

The expected risk can be bounded as:

$$\mathbb{E}[R(\theta)] \leq \hat{R}(\theta) + \Phi(\text{complexity of } f, l), \tag{1}$$

where $\hat{R}(\theta)$ is the empirical risk, and Φ is a function that increases with the complexity of f.

Theorem 1. *As the depth of a neural network increases, the risk of overfitting increases.*

Proof. Consider a neural network $f(x;\theta)$ with depth d. As d increases, the number of parameters and the expressiveness of f increase, leading to a higher VCdim($\mathcal{F}$), where $\mathcal{F}$ is the class of functions represented by the network. Thus, the second term in the expected risk bound (Equation 1) increases, indicating a higher risk of overfitting, especially when l (the size of the labeled set) is small. □

This proof demonstrates that deeper neural networks have a higher capacity, which can increase the risk of overfitting, particularly in semi-supervised learning scenarios with limited labeled data.